Jean Pierre Jacques Toussaint

Professor Brian

Southern New Hampshire University

CS-330-14033

April 17th, 2025

The 3D kitchen scene was built to closely mirror a real-world reference image, focusing on an authentic stovetop arrangement. Key objects like the stove with its burners and knobs, a cooking pot on one burner, and nearby jars and a bowl were chosen because they were prominent in the reference and defined the kitchen setting. Including these specific items ensures the virtual scene matches the real arrangement and feels immediately recognizable. To enhance realism and clarity, each model was given appropriate textures and material properties (for example, metal for the stove and pot and ceramic or glass for the bowl and jars), and a dynamic lighting setup was used to produce realistic highlights and shadows while keeping every element visible. These development choices result in a scene that closely reflects the reference image and makes all objects look lifelike and distinct.

The project implements an intuitive free-roaming camera so the user can fully explore the scene from any angle. Standard game-like controls were implemented: pressing W moves the camera forward, S moves backward, A strafes left, D strafes right, and the Q/E keys move the camera vertically down or up. Meanwhile, moving the mouse swivels the camera view direction, and the scroll wheel adjusts the zoom level (field of view), allowing both close-up inspection and wide-angle overviews. This combination of keyboard and mouse inputs lets the user smoothly navigate around the stove and surrounding objects. The camera setup provides dynamic viewing from multiple perspectives so that a user can easily inspect every detail of the 3D scene.

To manage complexity, the code is organized into custom helper functions that keep it modular and easy to maintain. Functions like SetTransformations handle all object positioning and scaling logic in one place. Similarly, SetShaderMaterial centralizes the setup of each object’s lighting and material properties in the shader, which avoids repetitive code for every model. Likewise, LoadSceneTextures was written to load and initialize all textures used in the scene so new textured objects can be added without duplicating the loading code. This modular design makes the code more readable, with each function having a clear purpose, and improves maintainability because changes (such as adjusting the lighting model or adding an object) can be made in one function rather than across many parts of the code.